Proceedings of the ASME 2016 Fluids Engineering Division Summer Meeting
FEDSM2016
July 10-14, 2016, Washington, DC, USA

DRAFT FEDSM2016-7853

ACCELERATING THE HIGH-FIDELITY SIMULATION OF TURBULENCE: ENSEMBLE AVERAGING

Vakhtang Makarashvili
Mathematics and Computer Science Division
Argonne National Laboratory
Lemont, IL, USA

Elia Merzari
Mathematics and Computer Science Division
Argonne National Laboratory
Lemont, IL, USA

Aleksandr Obabko
Mathematics and Computer Science Division
Argonne National Laboratory
Lemont, IL, USA

Paul Fischer
Mathematics and Computer Science Division
Argonne National Laboratory
Lemont, IL, USA

Andrew Siegel
Mathematics and Computer Science Division
Argonne National Laboratory
Lemont, IL, USA

ABSTRACT

Computational fluid dynamics (CFD) is increasingly used to simulate complex industrial systems. Most CFD analysis relies on the Reynolds-averaged Navier-Stokes (RANS) approach and traditional two-equation turbulence models. Higher-fidelity approaches to the simulation of turbulence such as wall-resolved large eddy simulation (LES) and direct numerical simulation (DNS) remain limited to smaller applications or to large supercomputing platforms. Nonetheless, continued advances in supercomputing are enabling the simulation of physical systems of increasing size and complexity. These simulations can be used to gain unprecedented insight into the physics of turbulence in complex flows and will become more widespread as petascale architectures become more accessible.

As the scale and size of LES and DNS simulations increase, however, the limitations of current algorithms become apparent. For larger systems, more temporal and spatial scale must be resolved, thus increasing the time-scale separation. While the smaller time scales dictate the size and the computational cost associated with each time step, the larger time scales dictate the length of the transient. An increased time-scale separation leads to smaller time steps and longer transients, eventually leading to simulations that are impractical or infeasible.

In practice the presence of multiple and strongly separated time scales limits the effectiveness of CFD algorithms for LES and DNS applied to large industrial systems. Moreover, the situation is likely to become worse on future computer architectures, as even larger systems will be simulated, thus increasing the size and length of transients. At the same time transients currently simulated on petascale architectures are unlikely to become any faster on exascale architectures.

In this paper we consider a technique to accelerate current transient simulations aimed at collect averaged turbulent statistics. The focus is on ergodic flows and simulations. This technique is ensemble averaging, commonplace in machine learning and artificial neural networks.

Ensemble averaging is the process of creating multiple models and combining them to produce a desired output. It is also at the basis of RANS/URANS turbulence modeling. In the proposed approach, multiple instances of the same ergodic flows are simulated in parallel for a short timeframe and summed to create an ensemble. Provided each instance is sufficiently statistically decorrelated, this allows considerable reduction in the time to solution.

This paper focuses on the theory and implementation of the methodology in Nek5000, a massively parallel open-source spectral element code. Moreover, we present the application of the method to the DNS and LES simulation of channel flow and pipe flow.

INTRODUCTION

Computational fluid dynamics (CFD) is increasingly used to simulate turbulent flows. Most CFD analysis, especially in industry, relies on the Reynolds-averaged Navier-Stokes (RANS) approach and traditional two-equation turbulence models. Higher-fidelity approaches to the simulation of turbulence, such as wall-resolved large eddy simulation (LES) and direct numerical simulation (DNS), remain limited to smaller applications or to large supercomputing platforms. In fact, since the Reynolds number dictates the local resolution, large machines are currently necessary to simulate engineering systems with turbulence-resolving techniques.

Nonetheless, continued advances in supercomputing are enabling the simulation of physical systems of increasing size and complexity. Current supercomputers can accommodate grids that reach tens to hundreds of billions of points, enabling the simulation of entire rod bundles with wall-resolved LES using CFD algorithms with good scalability properties.

As supercomputer become more powerful and larger, LES and DNS simulations become possible. With traditional algorithms, however, little can be done to make the solution of traditional problems run faster once the strong-scaling limit is reached. **Therefore, the time to solution of traditional algorithms is inherently limited.**

This problem will be particularly severe as computing platform move toward exascale. Because of power constraints, high-performance computing architectures are being designed to support extreme concurrency. Unfortunately, little can be done to reduce internode latency, which sets the node-level granularity of simulations and, ultimately, the rate at which work can be carried out. Extreme concurrency provides an avenue to solve larger problems rather than to solve today’s problems faster (assuming, as in the present case, that we are already running at the strong-scale limit). That will mean running cases orders of magnitude larger (at higher Reynolds numbers or for larger domain size) for longer integration times, as the time-scale separation will increase. Consequently, **accelerating turbulence calculations will become an imperative on larger architectures.**

In the present work we limit our attention to the ergodic flows. We seek to estimate expected values for the key quantities of interest in turbulent flows. The traditional approach is to exploit the ergodicity assumption and use time averaging as the basis of the estimate. But, as pointed out, statistical convergence is becoming increasingly slow given the eddy decorrelation timescale associated with large-scale engineering flows.

As an alternative, we propose and demonstrate here a direct **Ensemble Averaging** approach, whereby we carry out multiple statistically independent realizations of the flow and directly estimate the expected quantities using ensemble averaged statistics. This allows to employ efficiently larger processor counts.

We describe the method in the next section, we then describe the Nek5000 code and present some results for channel and pipe flow.

METHODS

We focus here on the solution of the constant-properties incompressible Navier-Stokes equations:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u}\nabla\vec{u} = -\frac{1}{\rho}\nabla p + \nu\Delta\vec{u} \qquad (1)$$

$$\nabla \cdot \vec{u} = 0 \qquad (2)$$

between times *t=0* and *t=T*. In ergodic turbulent flows equations (1) and (2) are highly unsteady, and the velocity and pressure signal are time dependent, while preserving time-independent statistical properties. Such properties may include the mean $\langle\vec{u}\rangle$ and the rms $\langle\overrightarrow{\acute{u}^2}\rangle$, as well as higher statistics (skewness, flatness). While several types of averaging have been applied in theory to estimate such properties [3], in practice the most common way to estimate mean and rms is to apply time-averaging:

$$\langle\vec{u}\rangle \approx \frac{1}{t1-t0}\int_{t0}^{t1}\vec{u}dt \qquad (3)$$

$$\langle\overrightarrow{\acute{u}^2}\rangle \approx \sqrt[2]{\frac{1}{t1-t0}\int_{t0}^{t1}(\vec{u}-\langle\vec{u}\rangle)^2 dt} \qquad (4)$$

and so on for higher statistics. Here t_0 and t_1 are respectively the initial time of collection of statistics (i.e., the time it reaches a statistical steady state) and the final time of collection. The latter is selected until further collection of statistics does not affect the result. This approach forces an additional time scale on the problem represented by (1) and (2) and given by $t_1 - t_0$. This in turn leads to a significant computational burden because this time scale can be large, often larger than the time required to reach a statistical steady state. Moreover, it may grow if higher statistics or turbulence budgets are of interest. The consequence is large typical time-to-solution values for turbulent flow simulations, which limit their practicality.

In addition to affecting the time to solution, the lack of time parallelism implies that, assuming simulations are already run at the strong-scaling limit and turbulent flow simulations are unable to employ effectively larger processor counts.

In this work we present an alternative method for estimating the constant-properties Navier-Stokes equations and other statistical properties. The method relies directly on the concept of ensemble and exploits finer granularity. Assuming *n* **independent** solutions $\vec{u}_n$ to the time-dependent problem (1–2), an estimator can be constructed:

$$\langle\vec{u}\rangle \approx \frac{1}{n}\frac{1}{s1-s0}\sum_{i=1}^{n}\int_{s0}^{s1}\overrightarrow{u_\imath}dt \qquad (5)$$

$$\langle\overrightarrow{\acute{u}^2}\rangle \approx \sqrt[2]{\frac{1}{n}\frac{1}{s1-s0}\sum_{i=1}^{n}\int_{t0}^{t1}(\overrightarrow{u_\imath}-\langle\vec{u}\rangle)^2 dt}, \qquad (6)$$

where s_0 and $s_1$are the beginning and end of the integration time, respectively, and are typically much smaller than t_0 and t_1. In order to find *n* independent solutions of (1–2), several strategies can be employed. The choice is critical, as we will discuss later, and will determine the resulting efficiency. As a proof of concept, however, we will here examine the most straightforward choice: Let each $\vec{u}_n$ be a solution of (1–2) starting from a different initial condition $\vec{u}_n^0$ at time *t=0*. This is clearly not optimal because it does not guarantee sufficient decorrelation a priori. However, we will demonstrate that even such a simple choice provides surprisingly powerful results. Each independent *i* solve of (1–2), with *i* from *1* to *n*, is referred to in the following as a **session**. Each session is completely independent of the others and can be performed on a set of disjoint processors with minimal communication. This allows to exploit larger processor counts effectively.

All methods have been implemented in the spectral-element code Nek5000. *We point out here that all sessions are carried out in a single run with minimal user involvement.*

NEK5000

The Argonne-based open-source fluid/thermal simulation code Nek5000 [4] is designed specifically for transitional and turbulent flows in complex domains. Nek5000 is based on the spectral-element method (SEM) [5], a high-order weighted residual technique that combines the geometric flexibility of finite elements with the rapid convergence and tensor-product efficiencies of global spectral methods. Globally, the SEM is based on a decomposition of the domain into E smaller subdomains (elements), which are assumed to be curvilinear hexahedra (bricks) that conform to the domain boundaries. Locally, functions within each element are expanded as *N*th-order polynomials cast in tensor-product form, which allows differential operators on N^3 gridpoints per element to be evaluated with only O(N^4) work and O(N^3) storage.

The principal advantage of the spectral-element method is that convergence is exponential in *N*, which implies that significantly fewer gridpoints per wavelength are required to accurately propagate a signal (or turbulent structure) over the extended times associated with high Reynolds number flow simulations. A high-order code involves slightly more work per gridpoint but not more memory access. We emphasize that the reduction in the number of gridpoints is a function of the discretization choice, independent of the implementation; the savings is realized by having an efficient SEM code that does not lead to increased cost per gridpoint.

In addition to its high-order foundation, Nek5000 has several other features. Temporal discretization is based on a high-order splitting that is third-order accurate in time and reduces the coupled velocity-pressure Stokes problem to four independent elliptic solves per timestep: one for each velocity component and one for the pressure. The velocity problems are diagonally dominant and thus easily solved by using Jacobi-preconditioned conjugate gradient iteration. The pressure substep requires a Poisson solve at each step, which is effected through multigrid-preconditioned GMRES iteration coupled with temporal projection to find an optimal initial guess. Particularly important components of Nek5000 are its scalable coarse-grid solvers that are central to parallel multigrid. The code features a fast direct solver that is optimal up to processor counts of P ≈ 10^4 and fast algebraic multigrid for P = 10^5 and beyond. Counts of 15 GMRES iterations per timestep for billion-gridpoint problems are typical with the current pressure solver.

Nek5000 scales extremely well on the BG/P and BG/Q architectures. The code realizes excellent strong scaling, sustaining 60% parallel efficiency with as few as 2,000 points per process for P=1,048,576. Typically, production runs are made with 5,000–10,000 points per process and thus run at higher parallel efficiencies.

The approach used to simulate turbulence in the present work is LES, with an explicit filter that mimics the de-convolution method [6]. In LES, large-scale turbulence is simulated while smaller scales are modeled. Since smaller scales have a nearly universal behavior, LES is a more reliable methodology than is Reynolds-averaged Navier-Stokes, in the sense that it generally depends less on the modeling assumptions. Nek5000 has been validated extensively both in DNS and LES mode [7, 8].

RESULTS

In the following, we present results for two canonical cases: turbulent channel flow and turbulent pipe flow. The results are compared with DNS data whenever possible [9]. The channel flow results are analyzed in more detail. The pipe flow results have the primary objective to show that the method is not limited to simplified geometries but has a broad appeal.

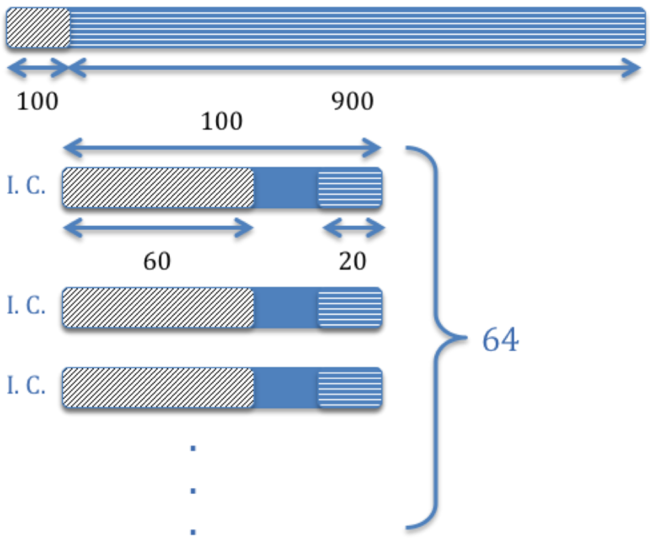

Fig. 1 Schematic of time domains for single-session and multisession runs for channel flow.

All simulation times in the manuscript are reported in convective time units: δ/U_b for channel flow and D/U_b for pipe flow.

Channel flow

Two sets of turbulent channel flow simulations were performed by using the high-order spectral-element code Nek5000. The first set was a single session run with 16 processors and a long simulation time (T = 1000) while the second set used 64 parallel sessions with a shorter time domain (T = 100). The goal was to employ an ensemble-averaging technique and achieve significant reduction in the time to solution with a computational cost comparable to that of the conventional time-average method.

Figure 1 illustrates the differences between the time-average and ensemble-average runs. The shaded area in the beginning of each block represents the time interval during which data was disregarded in order to wait until fully turbulent flow was developed and spatial velocity profiles across the sessions were decorrelated. Each session was started with similar but slightly different initial conditions adding random noise (i.e, using the session number *i* as a seed for the random number generator for instance).
The blue area with white horizontal lines shows the segment of the time domain where time-averaged data was collected. For the time-average run, this area was 90% of the entire time domain. Time-averaged velocity profiles were generated during this entire time interval. Data collection segments for the ensemble runs across the sessions constitute the last 20% of the time domain.

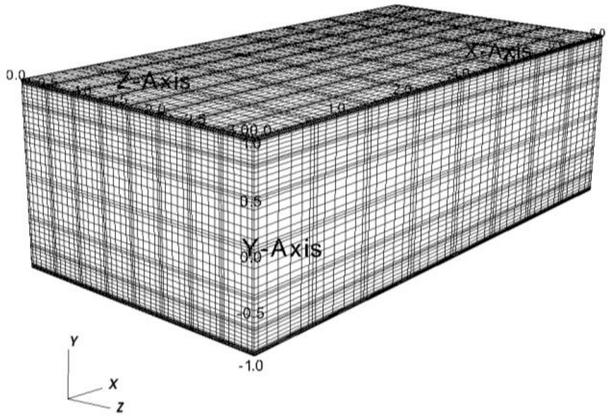

Fig. 2 Computational box for channel flow simulation with 512 elements. The size of the box is [2πδ x 2δ x πδ] in *x*, *y* and *z* directions, respectively. Dimensions normalized by δ.

A computational mesh used in channel flow simulations is given in Fig. 2. It consists of 512 (8x8x8) elements. The size of the computational box for both the time-averaged and ensemble-averaged runs was [2πδ x 2δ x πδ] in [*x y z*] directions, where δ is the channel half-width. The streamwise direction was along the *x*-axis. Periodic boundary conditions are applied in both the *x* and *z* directions. Both the time and ensemble runs of channel flow simulations were performed with polynomial order 9, for a total of approximatively 512,000 collocation points. A sensitivity study was performed to ensure this resolution was sufficient at the given polynomial order. The filtering for the deconvolution LES model was set to 5% on the last mode. In both cases the flow was driven by demanding a fixed bulk flow velocity – U_b, with Re_b = 10,000. The bulk Reynolds number is defined here as $Re = U_b\delta/\upsilon$.

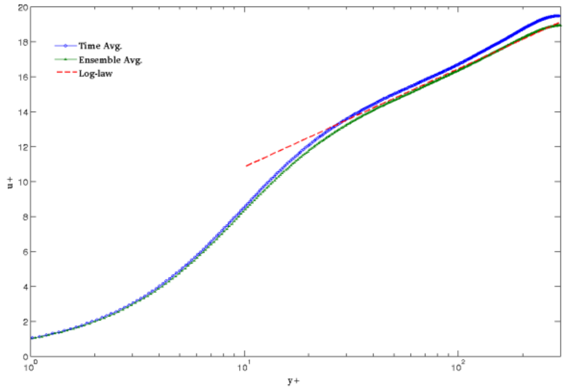

Fig. 3. Plot showing u+ as a function of y+ with time-averaged and ensemble-averaged data sets for turbulent channel flow. Re = 10,000, Re_τ = 300

Figure 3 shows u+ as a function of y+ for both data sets along with the log-law (red line) with constants k = 0.41 and B = 0.52. **Figure 4** illustrates <u^2> (scaled by friction velocity) obtained from the time-averaged and ensemble-averaged runs along with DNS results by Kasagi et al. [9] for channel flow at Re = 10,000. These plots were generated at the center of the *x* and *z* domain using 1,000 interpolation points in the *y* direction between the center of the channel and the wall ([0, δ]).

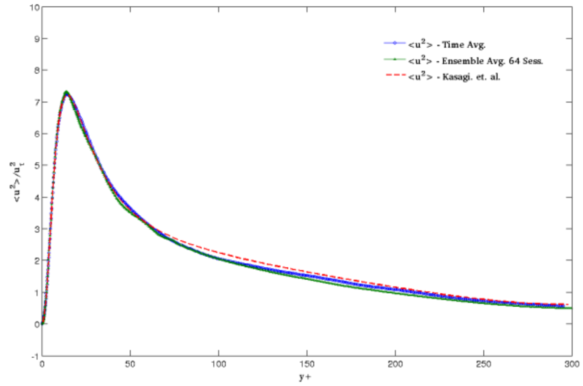

Fig. 4 Plot showing <u^2> (scaled by u^2_τ) with time-averaged and ensemble-averaged data along with DNS results [9]. Turbulent channel flow simulation at Re = 10,000 Re_τ = 300.

These plots demonstrate that the time and ensemble averages agree well with each other and also with theoretical and DNS results.

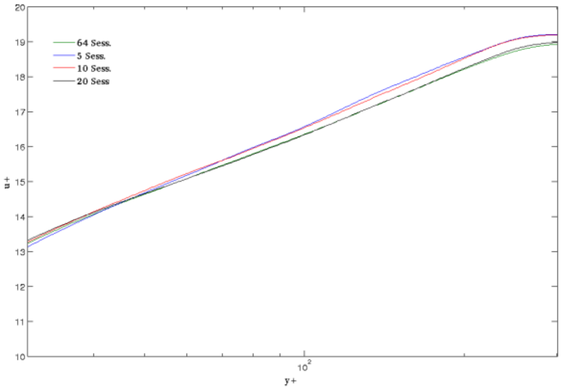

Fig. 5. Plot showing u+ as a function of y+ for ensemble-averaged data sets with different numbers of sessions zoomed in the log-law region to illustrate the convergence of results as the number of sessions increases. Re = 10,000.

The dependence of the u+ velocity profile on the number of sessions used in calculating averages was also examined, and results are presented in **Fig. 5**. The calculations were done for 5, 10, 20, and 64 sessions. The figure shows that the 20-session results are already close to convergence and produce satisfactory agreement with the 64-session results, which in turn has an excellent agreement with the log-law.

Table 1 summarizes the **c**omputational costs and time to solution for the time-averaged and ensemble-averaged cases with different numbers of sessions for channel flow . Since these simulations were performed on different platforms, a conversion factor was necessary to exclude the performance differences between different CPUs when comparing computation costs. This factor was determined based on wall time per time step in each case and turned out to be around 4.5 for channel flow simulations. The ensemble run was performed with 64 sessions and 128 CPUs (256 MPI ranks). However, in **Table 1** we also provide computational costs for the 45- and 20- session runs. We chose 45 because it represents the same total time domain as for the time-averaged case (45 x 20 => *T* = 900) and should have the same precision as the time-averaged results. The computational cost for 20 sessions was calculated because according to **Fig. 5** results from 20 sessions already agree well with the theoretical prediction. The table suggests that the ensemble-averaging technique tends to have a higher computational cost than does the time-averaging approach. The reason is that even after the turbulence is established, some of the statistics during the ensemble runs are disregarded (solid blue regions in the middle of the ensemble time domains in **Fig. 1**) in order to ensure decorrelation among the sessions. However, the 20-session ensemble run, which is close to the correct solution, has about the same computational cost as does the long time-averaged run. **We conclude that using ensemble runs significantly reduce the time to solution**.

This conclusion suggests that ensemble runs can utilize the full potential of a supercomputing platform, whereas the conventional time- averaged approach is subject to the strong-scaling limit and acquiring turbulence statistics for a long time domain often becomes unfeasible.

Table 1. Computational cost and time to solution for time-averaged and ensemble-averaged runs for channel flow.

	Computational Cost [cpu x hr]	Time to Solution [hr]
Time	16 x 4.5 x 36 = 2592	36
Ens. Actual	64 x 128 x 1 = 8192	1
Ens. Scaled	45 x 128 x 1 = 5760	1
Ens. 20 Sess.	20 x 128 x 1 = 2560	1

Pipe flow

A similar study was performed for pipe flow. **Figure 6** presents the differences between time-averaged and ensemble-averaged runs. Data was collected for the time-averaged run for *T* = 250 units and *T* = 80 for the ensemble average runs.

The first 50 units of time were needed to reach turbulent flow. The ensemble-averaged had slightly perturbed initial conditions based on the session number *i*. Only the data collected during last 10 time units was used for the collection of statistics. It is likely the time *T* could have been shortened in the ensemble averaging sessions, but due to time constraints that test was not performed here.

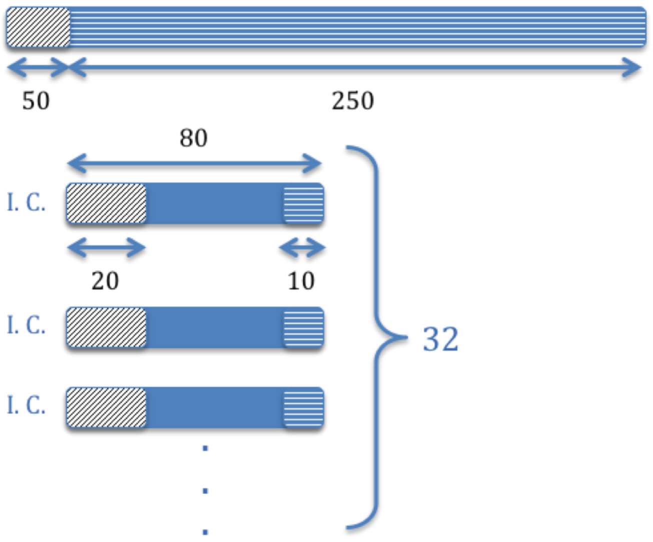

Fig. 6 Time domains for single session and multi session runs for pipe flow.

We emphasize that this particular calculation was carried out only for demonstration purposes. It is likely The ensemble run was performed with 32 sessions and 256 CPUs in each session (512 MPI ranks).

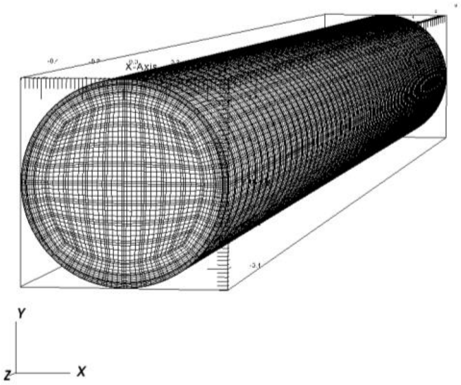

Fig. 7 Computational mesh for pipe flow simulation with 8,320 elements. The size of the box is *D* x *D* x 14*D* in *x*, *y*, and *z*, respectively. Dimensions normalized by *D*.

A refined computational mesh with 8,320 elements used for the pipe flow simulations is shown in **Fig. 7**. The filter assumed for the deconvolution LES model was 5% on the last mode of each element. The flow direction was along the *z*-axis. Periodic boundary conditions are applied in *z* and the length of the domain the streamwise direction is set to 14*D*. Both the time-averaged and ensemble-averaged runs were performed with polynomial order 5 for a total of less than 2 million collocation points. The results are slightly under-resolved result (Fig. 8-9) at this polynomial order and should be considered as such. The primary objective of this simulation is to demonstrate the method can be applied to complex geometries.

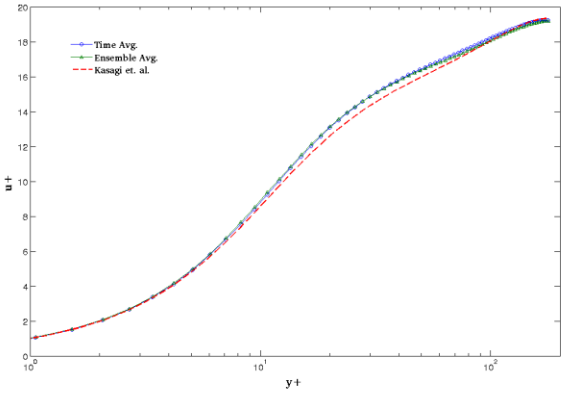

Fig. 8 Plot showing u+ as a function of y+ with time-averaged and ensemble-averaged data sets and DNS results [9] for turbulent pipe flow. Re = 5,310 Re_τ = 180.

Figure 8 and **Figure 9** present results for streamwise velocity and streamwise velocity rms profiles, respectively, in wall units for pipe flow simulations. Each plot includes time-averaged and ensemble-averaged results along with DNS results by Kasagi et. al. [9] for comparison. Both plots show that time-averaged and ensemble-averaged runs produce almost the same velocity profiles, and they both agree reasonably well with the DNS data given the slight under-resolution. The bulk Reynolds number $Re = U_b D/\nu$ under investigation is equal to 5,310.

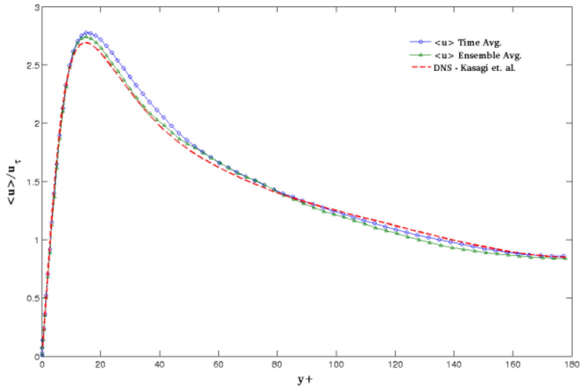

Fig. 9 Plot showing <uz> rms (scaled by u_τ) with time-averaged and ensemble-averaged data and DNS results [9] for turbulent pipe flow. Re = 5310 Re_τ = 180.

As in the case of the channel flow, computational costs and time-to-solution numbers were also calculated for pipe flow. The results are summarized in **Table 2**. Two cases are provided in the table for ensemble results: one for 32 sessions, which was the actual number that the simulation was run with, and the other for 25 sessions to represent the scaled case with the same total time domain (25 x 10 => *T* = 250) as the time-averaged run.

Table 2. Computational cost and time to solution for time-average and ensemble-average runs for pipe flow simulations.

	Computational Cost [cpu x hr]	Time to Solution [hr]
Time	16 x 3.2 x 28.5 = 1459	28.5
Ens. Actual	32 x 256 x 1 = 8192	1
Ens. Scaled	25 x 256 x 1 = 6400	1

The performance conversion factor for the two different platforms was around 3.2. The results indicate that the ratios of ensemble-averaged computational cost to the time-averaged computational cost are even higher than they were for the channel flow simulations. The reason is that the fraction of the time domain for the ensemble run during which statistics was recorded and used for calculations was 12.5% compared with 20% for the channel flow (Fig. 6). The ensemble-averaging technique applied to the turbulent pipe flow example also successfully demonstrated that this approach allows collecting high-quality statistics in just one hour. We point out that an analysis of the type carried out in Fig. 5 was not performed. Moreover, as already pointed out the ensemble average runs were too long and thus not optimal. Therefore the increase of computational cost observed here is likely due to a lack of optimization.

CONCLUSIONS

A novel technique for the computation of statistical properties of ergodic turbulent flows has been presented and demonstrated on two canonical flows. This technique, which relies directly on the concept of ensemble, exploits larger processor counts while maintaining the same granularity, by effectively parallelizing the statistics collection phase.

The results indicate that even with the most simplistic form of decorrelation, one can achieve massive speedups in terms of time to solution. Moreover, these speedups can be achieved with a minimal increase in computational cost in terms of CPU-hours if special care is taken in minimizing the number of sessions (Table 2).

While the main purpose of this paper is to provide a proof of concept, the method can be used to accelerate a collection of statistics in massive simulations where the strong-scaling limit has already been already reached.

Future work will focus on better characterization of the correlation between sessions. We will also optimize the process to decrease the lead time between the beginning of a session and the start of the data collection. Moreover the method will be demonstrated in additional geometries and more complex cases. Extension to non-ergodic flows will also be considered.

NOMENCLATURE

$\vec{\mathbf{u}}$	Perturbation velocity
x, y, z	Cartesian coordinates
$\boldsymbol{\upsilon}$	Kinematic viscosity
$\boldsymbol{\rho}$	Density
$\mathbf{p}$	Pressure
<,>	Ensemble averaging operator
$\boldsymbol{u_i}$	Independent *i* realization
$\boldsymbol{T}$	Total time
$\mathbf{U_b}$	Bulk velocity
$\mathbf{D}$	Diameter
$\boldsymbol{\delta}$	Channel half-width
$\boldsymbol{Re}$	Bulk Reynolds number
$\boldsymbol{Re_\tau}$	Friction Reynolds number
$\boldsymbol{u_\tau}$	Friction velocity
y+	Distance from the wall in wall units
u+	Streamwise velocity in wall units

ACKNOWLEDGMENTS

This material was based upon work supported by the U.S. Department of Energy, Office of Science, as part of the CESAR Program, under contract No. DE-AC02-06CH11357.

REFERENCES

[1] Elia Merzari, Aleks Obabko, Paul Fischer, Noahord, Justin Walker, Andrew Siegel, "LARGE-SCALE SIMULATION OF NUCLEAR REACTORS: ISSUES AND PERSPECTIVES", *Proceedings of NURETH-16*, Chicago, IL, August 31st-September 4th 2015

[2] U. Naftaly, N. Intrator, and D. Horn. "Optimal ensemble averaging of neural networks" , *Network: Computation in Neural Systems*, **Vol. 8**, pp. 283-296 (1997)

[3] S. B. Pope. Turbulent flows. In Cambridge University Press (2000)

[4] Fischer, P., Lottes, J., Pointer, W.D., and Siegel, A., “Petascale Algorithms algorithms for Reactor reactor Hydrodynamicshydrodynamics,” *J. Phys. Conf. Series*, **125**, no. 1. (2008)

[5] M. Deville, P. Fischer, and E. Mund, High-Order Methods for Incompressible Fluid Flow, Cambridge University Press, Cambridge, UK, 2002.

[6] S. Stolz, P. Schlatter and L. Kleiser, ”High-pass filtered eddy-viscosity models for large-eddy simulations of transitional and turbulent flow”, *Phys. Fluids* **17**, 065103 (2005)

[7] J. Walker, E. Merzari, A. Obabko, P. Fischer and A. Siegel, “Accurate Prediction of the Wall Shear Stress in Rod Bundles with the Spectral Element Method at High Reynolds Numbers”, *International Journal of Heat and Fluid Flow* , **50**, Pages 287–299 (2014)

[8] E. Merzari, P. Fischer, W.D. Pointer, “ Numerical Simulation and Proper Orthogonal Decomposition of the flow in a Counter‐Flow T‐junction.” *Journal Fluid Engineering*, **135**, 091304, (2013)

[9] Kasagi et al., “DNS DATABASE OF TURBULENCE AND HEAT TRANSFER”, http://thtlab.jp/index-orig.html (2012)

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.